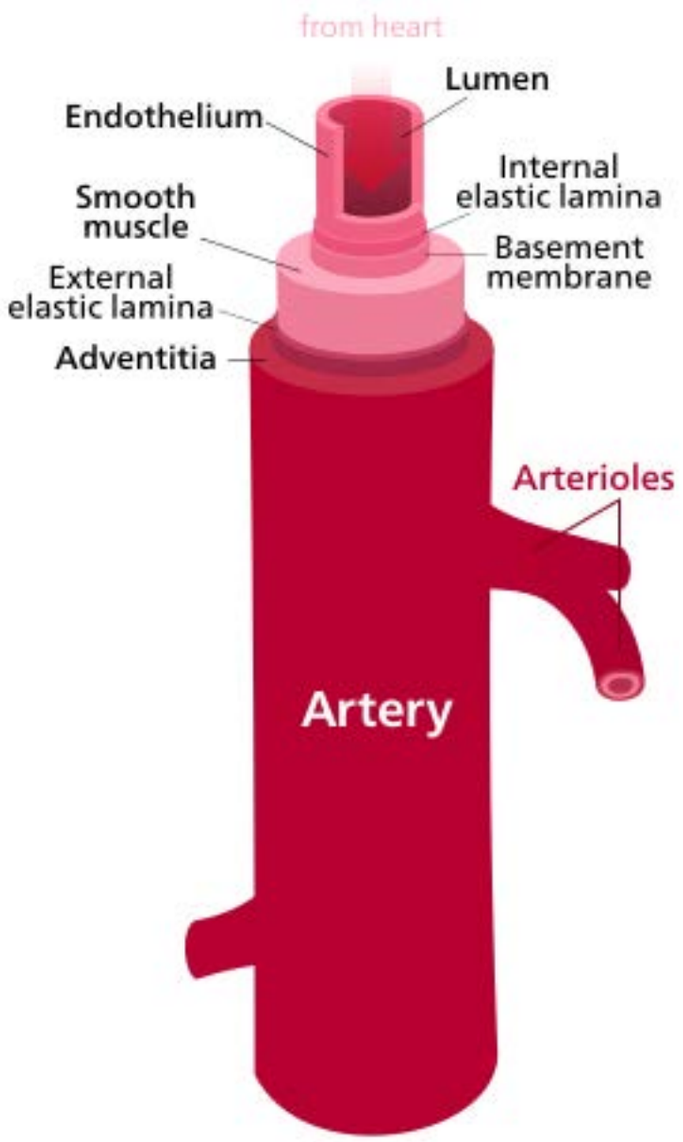
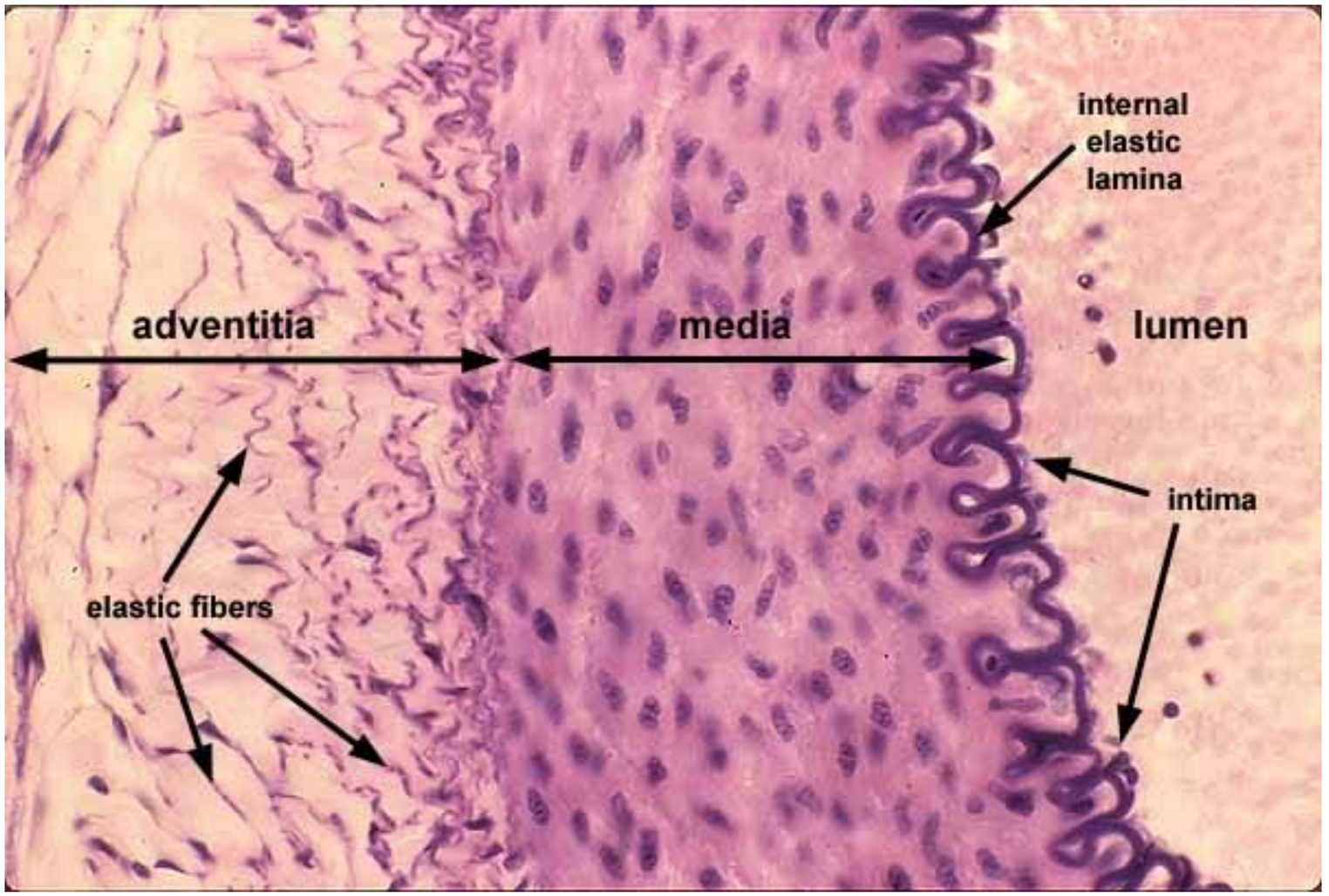
# Efficient computing in evolution equations with memory

Paris Perdikaris
Massachusetts Institute of Technology, Department of Mechanical Engineering
***Web:*** `http://web.mit.edu/parisp/www/`
***Email:*** `parisp@mit.edu`

Crunch group FPDE club
August 12, 2016

# Why fractional equations?…*a motivating example*



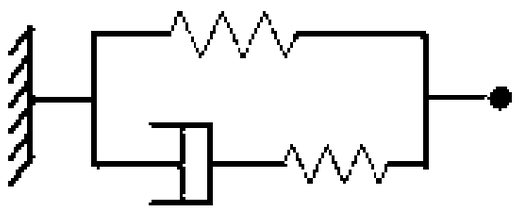

***Soft tissue biomechanics:***

- Complex biomaterials with non-trivial properties
- Non-linear elastic response
- Viscoelastic creep and continuous relaxation time-scales

Fung, Yuan-cheng. *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media, 2013.

# Why fractional equations?…*a motivating example*

***Modeling approach #1:*** Simple parametric integer-order models

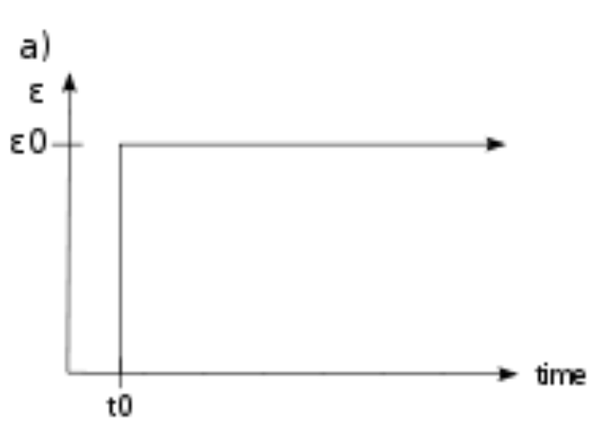


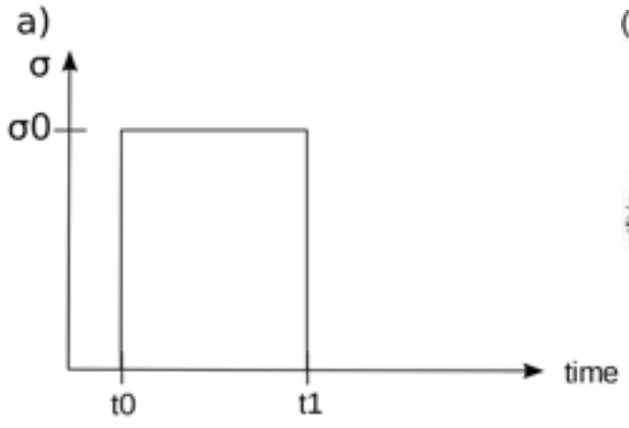
Standard Linear Solid model (SLS):

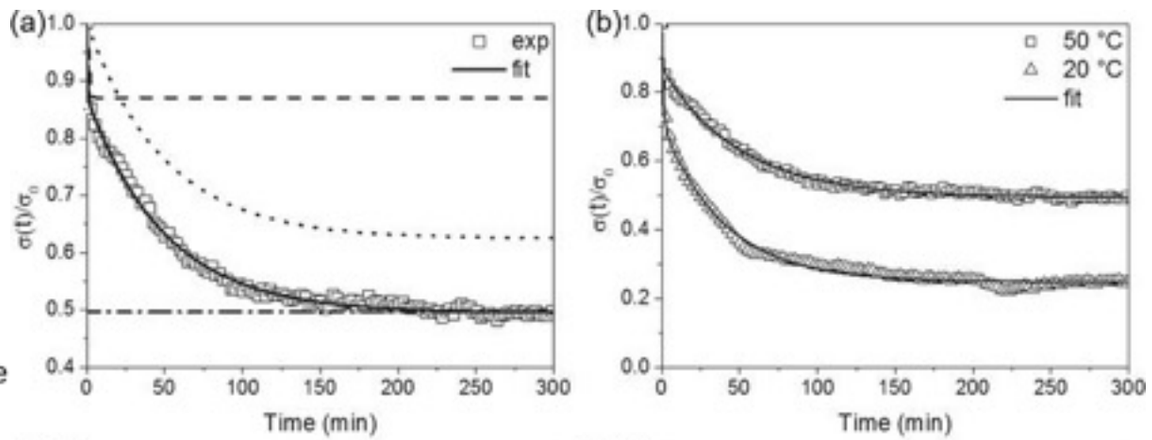
$$\sigma(t) + \tau_\sigma \frac{d\sigma(t)}{dt} = E[\epsilon(t) + \tau_\epsilon \frac{d\epsilon(t)}{dt}]$$

***3 parameters***

- Linear elastic response
- Simplest model that accounts for viscoelastic creep and relaxation
- Two discreet relaxation time-scales



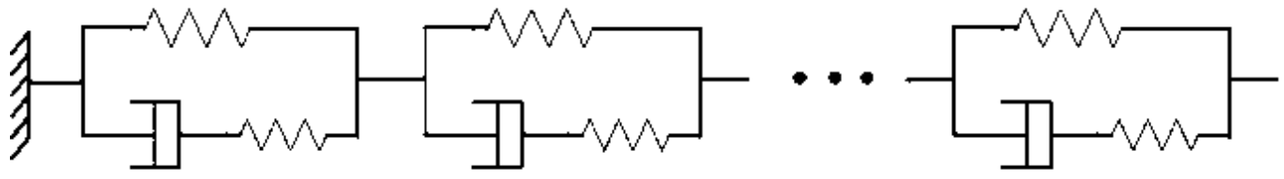



# Why fractional equations?…*a motivating example*

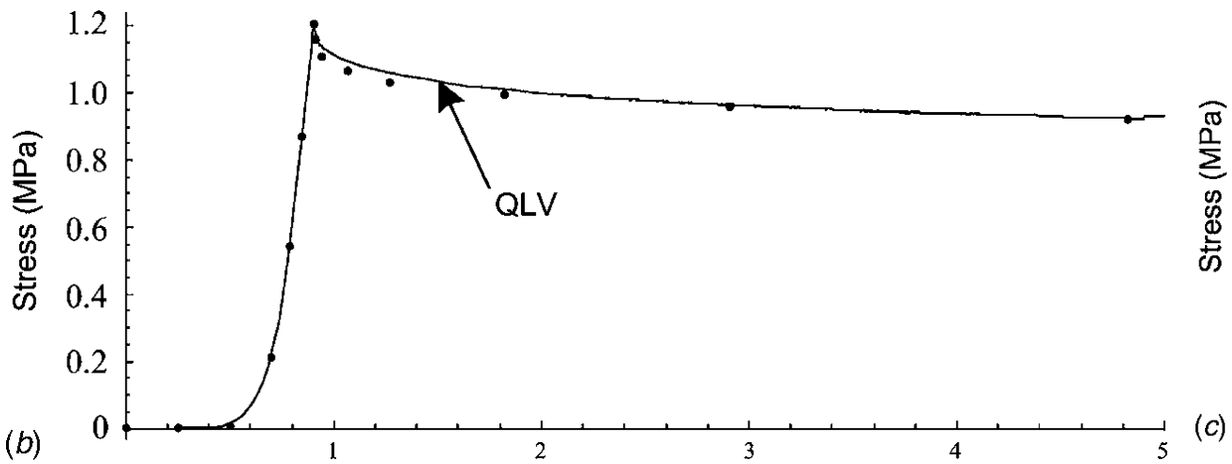
***Modeling approach #2:*** More complex parametric integer-order models

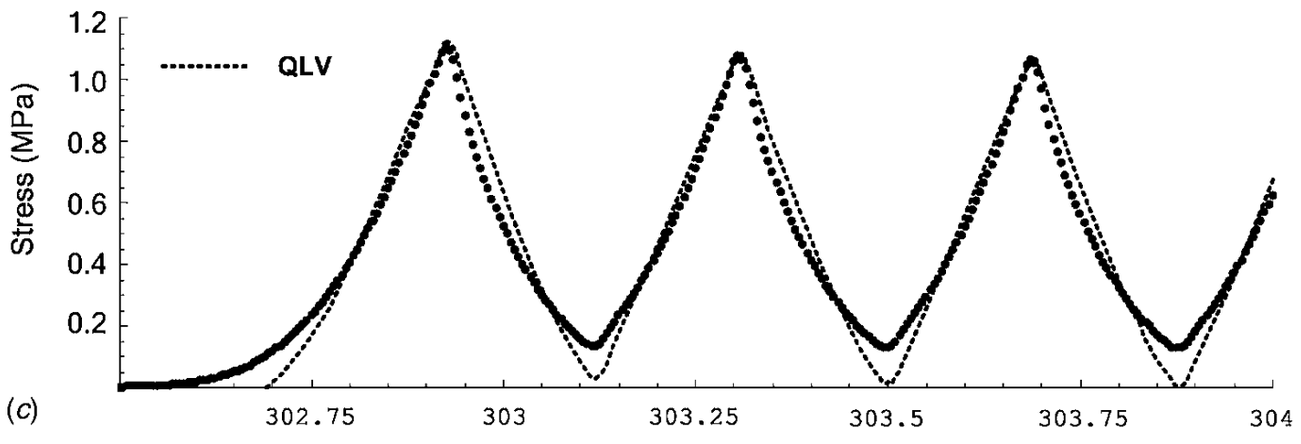
QLV Kelvin-Zener solids in series



***O(10) parameters***

- Linear elastic response
- Can capture more complex viscoelastic creep and relaxation behavior
- Multiple discreet relaxation time-scales




- Many parameters to be estimated from few data
- High parametric sensitivity
- The response behaves like a stochastic process and depends on patient-specific factors, age, etc.

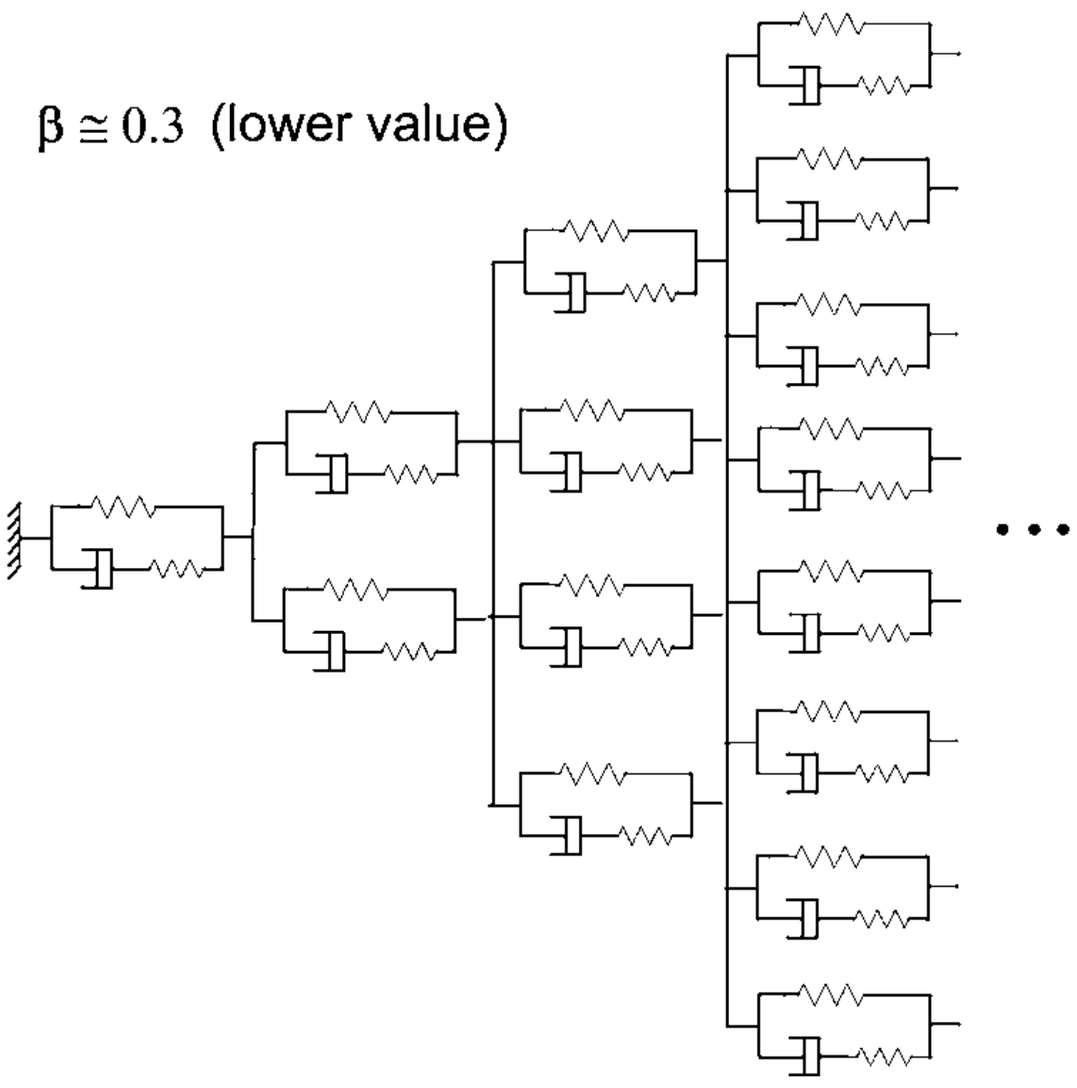
Fung, Yuan-cheng. *Biomechanics: mechanical properties of living tissues*. Springer Science & Business Media, 2013.

# Why fractional equations?…*a motivating example*

***Modeling approach #3:*** Even more complex parametric integer-order models

QLV Kelvin-Zener solids in series

$\beta \cong 0.3$ (lower value)



- Linear elastic response
- Can capture even more complex viscoelastic creep and relaxation behavior
- Continuous spectrum of relaxation time-scales

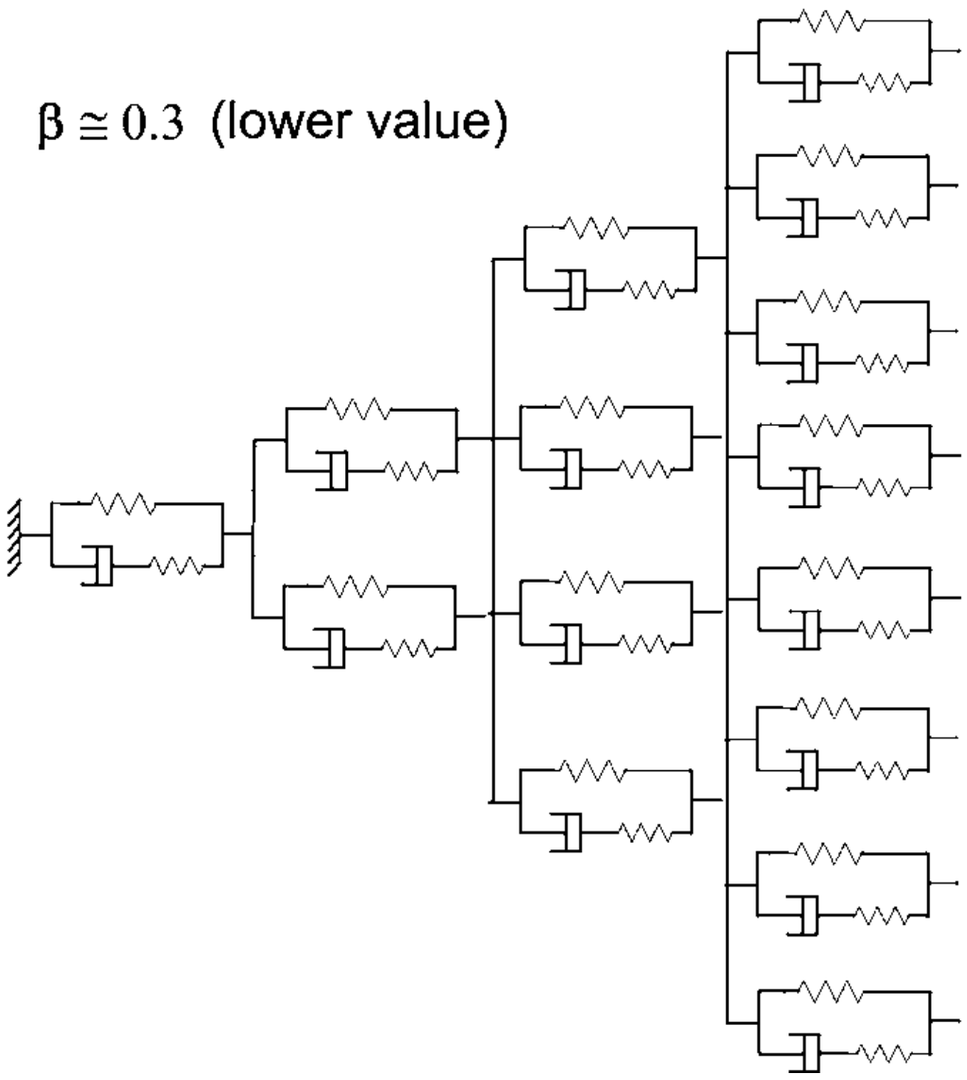
***…Guess how many parameters?***

# Why fractional equations?…*a motivating example*

***Modeling approach #4:*** Fractional-order models

Fractional-order Kelvin-Zener solid

$\beta \cong 0.3$ (lower value)



…

$$\sigma(t) + \tau_{\sigma\,0}^{\alpha C} D_t^{\alpha} \sigma(t) = E\left[\epsilon(t) + \tau_{\epsilon\,0}^{\alpha C} D_t^{\alpha} \epsilon(t)\right]$$

- Linear elastic response
- Can capture even more complex viscoelastic creep and relaxation behavior
- Continuous spectrum of relaxation time-scales

***4 parameters!***

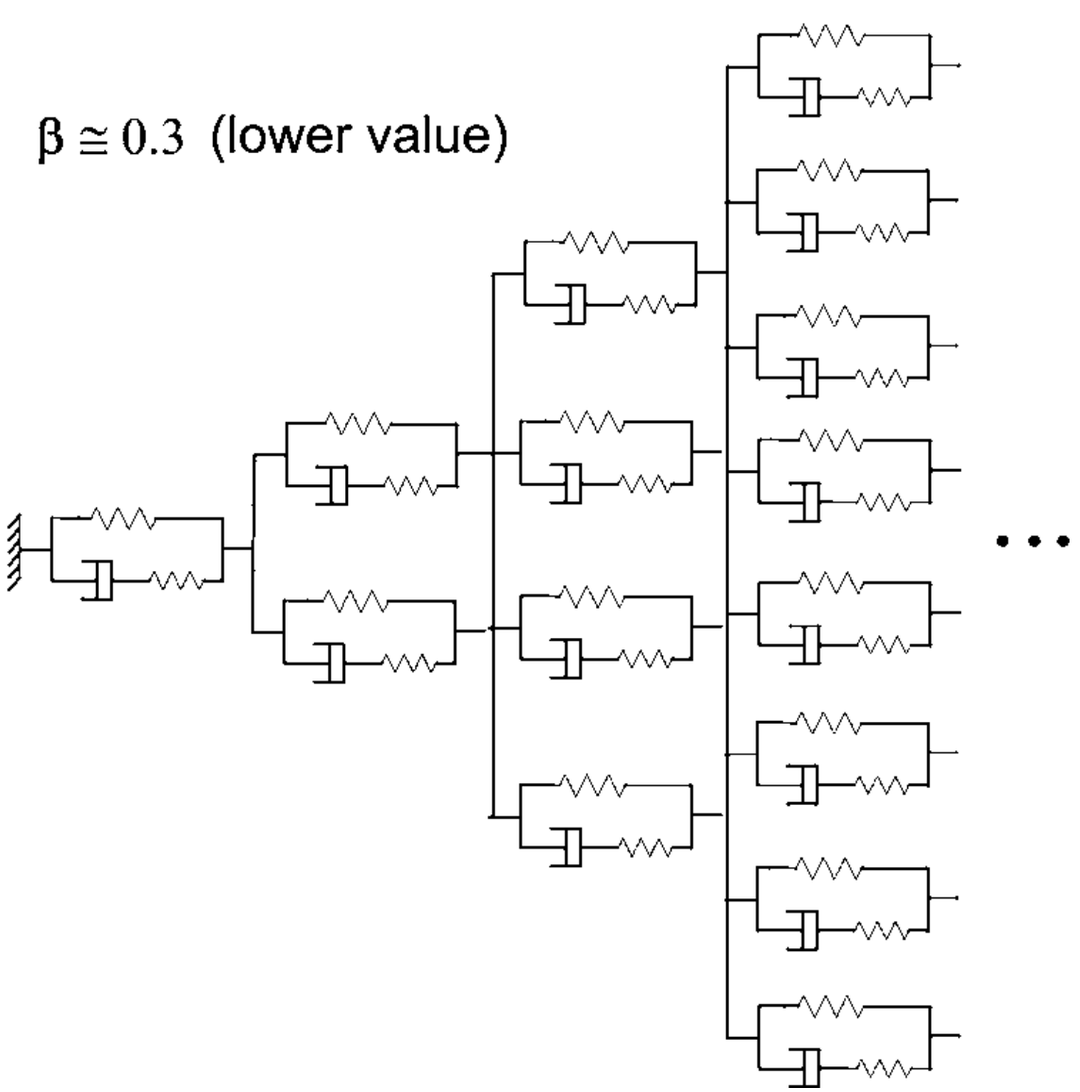
*Doehring, Todd C., et al. "Fractional order viscoelasticity of the aortic valve cusp: an alternative to quasilinear viscoelasticity." Journal of biomechanical engineering 127.4 (2005): 700-708.*

*Näsholm, Sven Peter, and Sverre Holm. "On a fractional Zener elastic wave equation." Fractional Calculus and Applied Analysis 16.1 (2013): 26-50.*

# Why fractional equations?…*a motivating example*

***Modeling approach #4:*** More complex fractional-order models

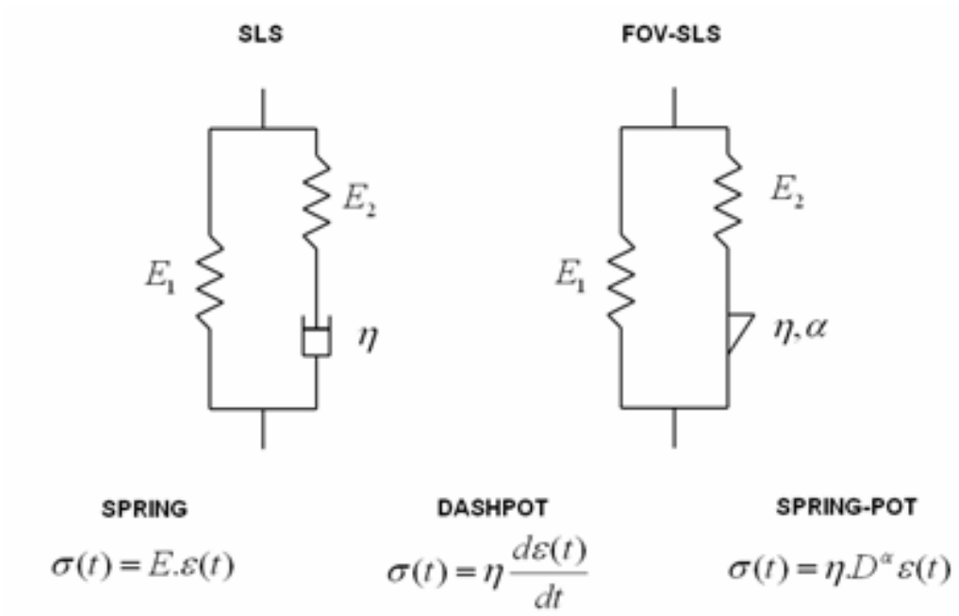
Fractional-order Kelvin-Zener solid




- Linear elastic response
- Can capture even more complex viscoelastic creep and relaxation behavior
- Continuous spectrum of relaxation time-scales

$$\sigma(t) + \tau_{\sigma\,0}^{\alpha C} D_t^{\alpha} \sigma(t) = E\left[\epsilon(t) + \tau_{\epsilon\,0}^{\alpha C} D_t^{\alpha} \epsilon(t)\right]$$

***4 parameters!***

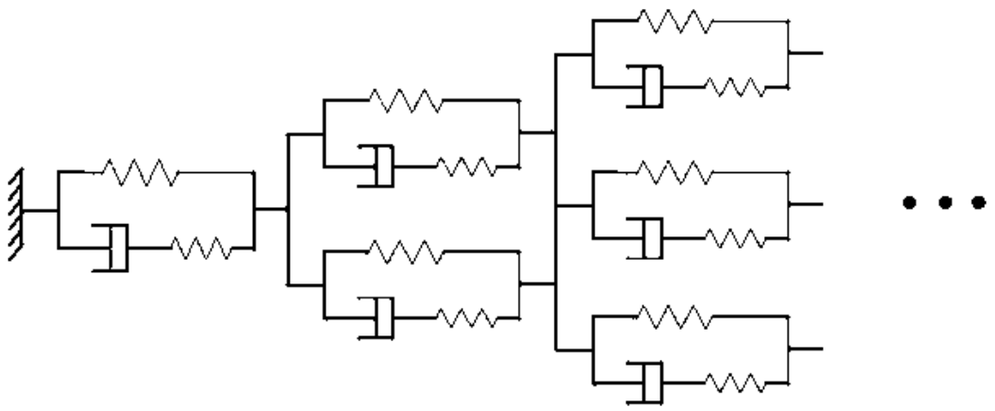



***…fewer parameters to be estimated, hence less parametric sensitivity.***

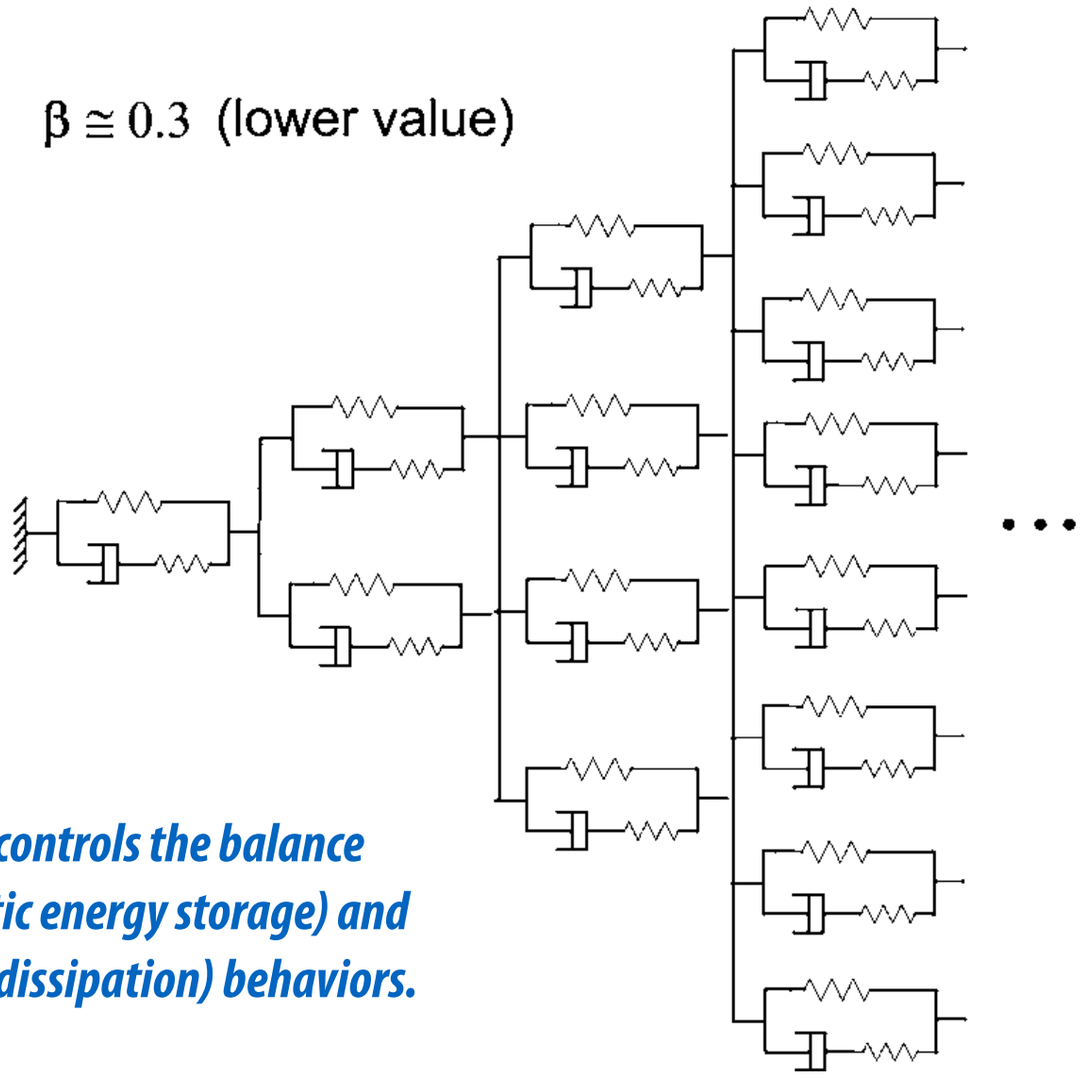
***…combine fractional solids in series and you can fit an elephant in the room (e.g. smooth muscle activation in cerebral auto regulation).***

# The role of the fractional order

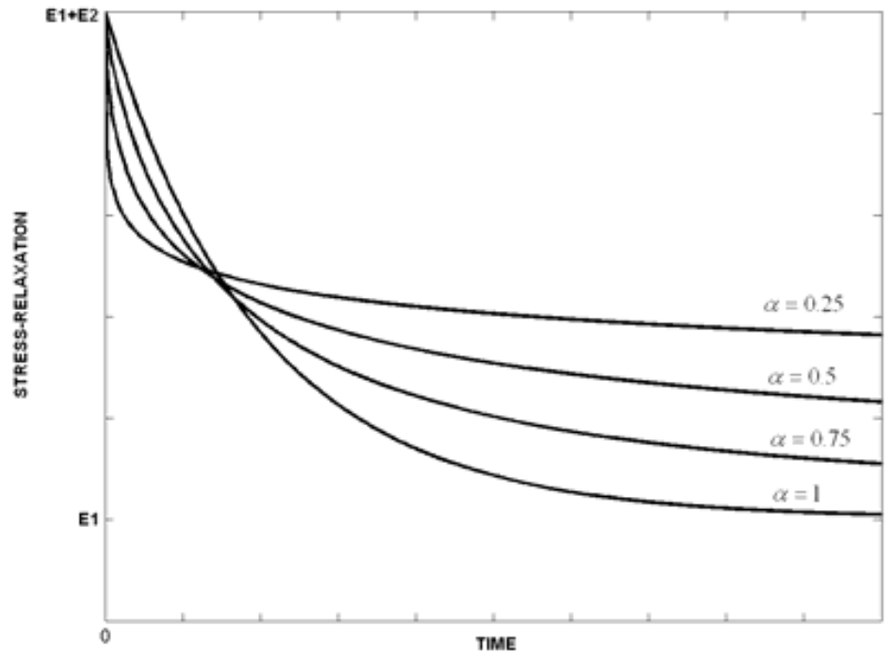
$\beta \cong 0.7$ (higher value)

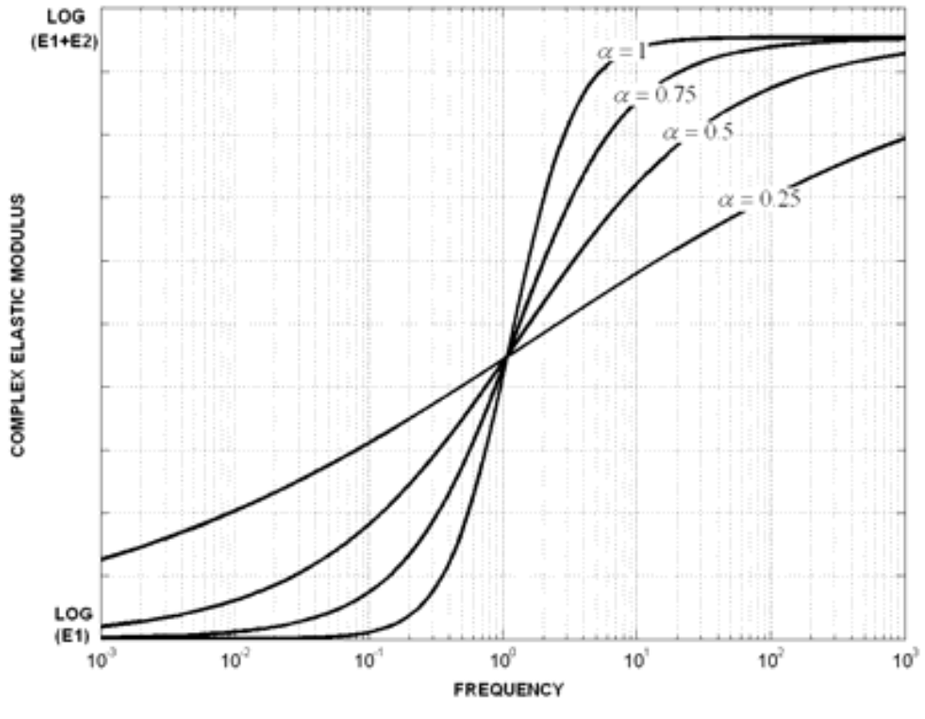


$\beta \cong 0.3$ (lower value)



***The fractional order effectively controls the balance between the conservative (elastic energy storage) and dissipative (viscoelastic energy dissipation) behaviors.***



*Great, but how do we compute now?*

# Discretize the derivatives

Grunwald-Letnikov approximation

$$ {}_0^C D_t^\alpha f(t) = \lim_{\Delta t \to 0} \Delta t^{-\alpha} \sum_{k=0}^{\infty} GL_k^\alpha f(t - k\Delta t), \quad GL_k^a := \frac{k-\alpha-1}{k} GL_{k-1}^\alpha $$

$$ \vdots $$

$$ p(x,t) = p_{ext} + \frac{1+\tau_\epsilon^\alpha \Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} p^E(x,t) + \frac{\Delta t^{-\alpha}}{1+\tau_\sigma^\alpha \Delta t^{-\alpha}} \sum_{k=0}^{\infty} GL_k^\alpha \{\tau_\epsilon^\alpha p^E(t-k\Delta t) - \tau_\sigma^\alpha p(t-k\Delta t)\} $$

- Low accuracy
- Excessive memory requirements, especially for advection dominated problems due to the CFL condition

***…why should we discretize in the first place?***
***These are linear time-fractional evolution equations with well behaved forcing terms…***
***…we can use our good old Laplace transforms and solve them analytically!***

# Computing convolution integrals

$$p(x,t) = p_{ext} + p^E(x,t) + p^V(x,t)$$

$$p^E(x,t) = \left(\frac{\tau_\epsilon}{\tau_\sigma}\right)^\alpha \beta(\sqrt{A} - \sqrt{A_0})$$

$$p^V(x,t) = \frac{1}{\tau_\sigma}\left(1 - \left[\frac{\tau_\epsilon}{\tau_\sigma}\right]^\alpha\right) \int_0^t E_{\alpha,0}\left(-\left[\frac{t-\gamma}{\tau_\sigma}\right]^\alpha\right) p^E(\gamma) d\gamma$$

| | | |
|---|---|---|
| ***Cost of a naive implementation:*** | $O(N^2)$ | operations |
| | $O(N)$ | evaluations of the kernel (e.g. Mittal-Leffler function) |
| | $O(N)$ | active memory |

***Should be more accurate than the GL discretization but it doesn't buy us much in terms of computational efficiency.***

# Fast convolutions for evolution equations with memory

Analytical solution of FPDEs using Laplace transforms gives rise to convolution integrals:

$$\int_0^t f(t-\tau)\, g(\tau)\, d\tau\,, \qquad 0 \le t \le T$$

The fast convolution method of *[1]* allows to compute such integrals with spectral accuracy and

- $O(N \log N)$ operations,
- $O(\log N)$ evaluations of the Laplace transform $F = \mathcal{L}f$, none of $f$, and
- $O(\log N)$ active memory.

vs

| | |
|---|---|
| $O(N^2)$ | for a naive |
| $O(N)$ | quadrature |
| $O(N)$ | implementation |

***Example:*** In fractional-order viscoelastic constitute laws we need to compute a memory term:

$$\int_0^t E_{\alpha,0}\left(-\left[\frac{t-\gamma}{\tau_\sigma}\right]^\alpha\right) p^e(\gamma) d\gamma$$

The fast convolution method bypasses costly evaluations of the Mittag-Leffler kernel: $E_\alpha(x) = \sum_{j=0}^{\infty} \frac{x^j}{\Gamma(1+\alpha j)}$

..and only requires simple function evaluations in the Laplace domain: $F(s) = \frac{1}{1+s^\alpha}$

*[1] López-Fernández, María, Christian Lubich, and Achim Schädle. "Adaptive, fast, and oblivious convolution in evolution equations with memory." SIAM Journal on Scientific Computing 30.2 (2008): 1015-1037.*

# Implementation

$$\int_a^b f(t-\tau)\,g(\tau)\,d\tau = \int_a^b \frac{1}{2\pi i}\int_\Gamma F(\lambda)\,e^{(t-\tau)\lambda}\,d\lambda\;g(\tau)\,d\tau$$
$$= \frac{1}{2\pi i}\int_\Gamma F(\lambda)\,e^{(t-b)\lambda}\int_a^b e^{(b-\tau)\lambda}\,g(\tau)\,d\tau\;d\lambda \approx \int_a^b \sum_{j=-N}^{N} w_j\,F(\lambda_j)\,e^{(t-\tau)\lambda_j}\,g(\tau)\,d\tau \;=\; \sum_{j=-N}^{N} w_j\,F(\lambda_j)\,e^{(t-b)\lambda_j}\,y(b,a,\lambda_j)$$

where the inner integral, henceforth denoted by $y(b,a,\lambda)$, is recognized as the solution at time $b$ of the scalar linear initial value problem

$$y' = \lambda y + g\ , \qquad y(a) = 0.$$

The $2N+1$ differential equations (2.5) with $\lambda = \lambda_j$ are solved approximately by replacing the function $g$ with its piecewise linear approximation and then solving exactly. Setting $g_n = g(a+n\Delta t)$, we get approximations $y_n \approx y(a+n\Delta t)$ recursively via

$$y_{n+1} = e^{\Delta t\lambda}\,y_n + h\int_0^1 e^{(1-\theta)\Delta t\lambda}\left(\theta g_{n+1} + (1-\theta)g_n\right)d\theta$$
$$= y_n + \frac{e^{\Delta t\lambda}-1}{\Delta t\lambda}\left(\Delta t\lambda y_n + \Delta t g_n + \Delta t\,\frac{g_{n+1}-g_n}{\Delta t\lambda}\right) - \Delta t\,\frac{g_{n+1}-g_n}{\Delta t\lambda}. \tag{2.7}$$

To estimate the error, note that in total we approximate

$$\int_a^b f(t-\tau)\,g(\tau)\,d\tau \approx \int_a^b \widetilde{f}(t-\tau)\,\widetilde{g}(\tau)\,d\tau,$$

where $\widetilde{f}$ is the quadrature approximation to $f$ constructed in the previous subsection, whose error is well under control, and $\widetilde{g}$ is the piecewise linear interpolant of $g$. (Higher-order interpolants of $g$ might also be used, but we have not implemented such an extension.)

# Implementation

Laplace transform

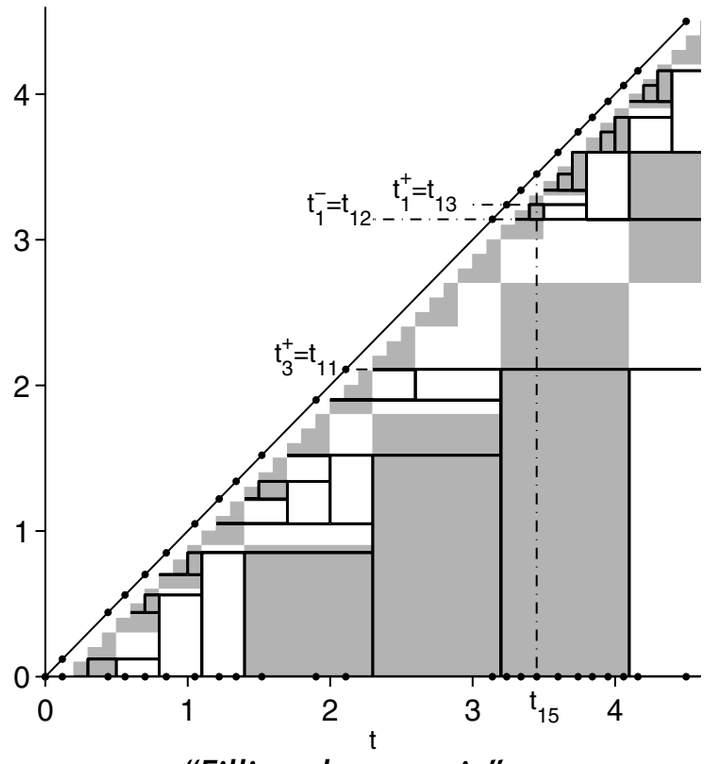
$$\int_0^{t_n} f(t_n-\tau)g(\tau)\,d\tau = \int_0^{t_n} \frac{1}{2\pi i}\int_\Gamma e^{(t_n-\tau)\lambda}F(\lambda)\,d\lambda\,g(\tau)\,d\tau$$

$$\int_{t^-}^{t^+} f(t_n-\tau)g(\tau)\,d\tau = \int_{t^-}^{t^+} \frac{1}{2\pi i}\int_\Gamma e^{(t_n-\tau)\lambda}F(\lambda)\,d\lambda\,g(\tau)\,d\tau$$

$$\approx \int_{t^-}^{t^+} \sum_{k=-K}^{K} w_k\, e^{(t_n-\tau)\lambda_k}\, F(\lambda_k)\, g(\tau)\,d\tau$$

$$= \sum_{k=-K}^{K} w_k\, F(\lambda_k)\, e^{(t_n-t^+)\lambda_k} \int_{t^-}^{t^+} e^{(t^+-\tau)\lambda_k} g(\tau)\,d\tau$$

$$= \sum_{k=-K}^{K} w_k\, F(\lambda_k)\, e^{(t_n-t^+)\lambda_k}\, y(t^+,t^-,\lambda_k),$$




*"Filling the mosaic"*

weights $w_k$ and quadrature nodes $\lambda_k$ given by

$$w_k = \frac{\tau}{2\pi i}\,\gamma'(k\tau)\ , \quad \lambda_k = \gamma(k\tau)$$

where $y(t^+,t^-,\lambda_k)$ is the solution at $t^+$ to the linear inhomogeneous ODE $y' = \lambda_k y + g, \quad y(t^-) = 0, \qquad -K \le k \le K$

Forward Euler, Runge-Kutta, BDF, etc

***Key ingredients:***

- Complex plane quadrature (Talbot contours)
- Local reduction to ODEs
- Adaptivity and smart memory management

[1] López-Fernández, María, Christian Lubich, and Achim Schädle. "Adaptive, fast, and oblivious convolution in evolution equations with memory." SIAM Journal on Scientific Computing 30.2 (2008): 1015-1037.

## *Matlab demo*

$$\int_a^b f(t-\tau)d\tau$$

$$f(t) = \frac{1}{\sqrt{\pi t}} \qquad \text{for which} \qquad F(s) = s^{-1/2}$$